ARTIKEL PEMBAHASAN TENTANG ENCAPSULATION PEMROGRAMAN JAVA

Disusun Oleh :

Adam Hendrawan[1] , Ahmad Panji[2] Setyawan dan Yuniana[3] Cahyaningrum

Pendidikan Teknologi Informasi,IKIP PGRI Bojonegoro[123].

Email:

adam.hendrawan04@gmail.com[1], panjisetyawan026@gmail.com[2], yuniana@ikippgribojonegoro[3]

Abstrak

Enkapsulasi merupakan konsep dalam pemrograman berorientasi objek (OOP) yang memungkinkan pengguna untuk menyembunyikan detail implementasi dari suatu kelas dari pengguna lain. Konsep ini bertujuan untuk memperbaiki keamanan dan perbaikan program, serta memudahkan pemeliharaan kode.Dalam enkapsulasi, variabel dan method dalam suatu kelas didefinisikan dengan access modifier seperti private atau protected, sehingga hanya dapat diakses melalui method yang telah ditentukan (getter dan setter). Dengan demikian, variabel dan metode tersebut tidak dapat diubah secara langsung dari luar kelas, sehingga memperbaiki program keamanan. Selain itu, enkapsulasi juga memudahkan pemeliharaan kode karena pengguna dapat mengubah implementasi suatu kelas tanpa mempengaruhi kelas lain yang menggunakan kelas tersebut.

Kata kunci: Pengertian Enkapsulasi, pribadi,dan terlindung.

I. PENDAHULUAN

Enkapsulasi adalah konsep dalam pemrograman berorientasi objek (OOP) yang pertama kali diperkenalkan oleh bahasa pemrograman Simula pada tahun 1960-an. Konsep ini kemudian diadopsi oleh bahasa pemrograman seperti Smalltalk, C++, dan Java. Enkapsulasi memungkinkan pengembang perangkat lunak untuk menyembunyikan implementasi detail suatu kelas dari pengguna lain.

Dalam enkapsulasi, variabel dan metode dalam suatu kelas didefinisikan dengan access modifier seperti private atau protected. Hal ini membuat variabel dan metode hanya dapat diakses melalui metode yang telah ditentukan (pengambil dan penyetel). Dengan demikian, detail implementasi suatu kelas tidak dapat diubah secara langsung dari luar kelas, meningkatkan program keamanan.

Selain itu, enkapsulasi juga memudahkan pemeliharaan kode. Pengguna dapat mengubah implementasi suatu kelas tanpa mempengaruhi kelas lain yang menggunakannya. Detail kelas implementasi terlarang, sehingga pengguna hanya perlu berinteraksi dengan metode-metode yang telah ditentukan. Perubahan pada kelas implementasi tidak memerlukan perubahan pada kode pengguna, mempermudah pemeliharaan dan pengembangan program secara keseluruhan.

Dalam artikel ini, kita akan membahas lebih lanjut tentang konsep enkapsulasi dalam pemrograman Java dan bagaimana enkapsulasi dapat meningkatkan keamanan dan pemeliharaan kode.

II. TINJAUAN PUSTAKA

Enkapsulasi dalam bahasa pemrograman Java adalah untuk menyembunyikan detail implementasi dari penggunaan objek, sehingga objek dapat digunakan dengan lebih mudah dan aman. Enkapsulasi dapat

diimplementasikan melalui penggunaan modifier access control seperti private, protected, dan public.

Dalam buku "Java A Beginner's Guide, Seventh Edition" oleh Herbert Schildt, enkapsulasi dijelaskan sebagai berikut:

"Enkapsulasi adalah konsep kunci dalam pemrograman yang berorientasi objek. Dalam bahasa Java, enkapsulasi diterapkan melalui penggunaan pengubah kontrol akses. Ketika sebuah kelas didefinisikan, data dan metode yang memanipulasi data tersebut dapat ditetapkan sebagai publik, dilindungi, atau pribadi. Ketika data atau metode diberi pengubah private, hanya class itu sendiri yang dapat mengakses data atau metode tersebut."[1]

III. METODE

Dalam artikel ini, metode yang digunakan adalah refleksi pustaka untuk memperoleh pemahaman yang mendalam tentang konsep enkapsulasi dalam pemrograman yang berorientasi objek. Tinjauan pustaka dilakukan dengan mencari dan menganalisis sumber literatur yang relevan, seperti buku, jurnal ilmiah, dan sumber-sumber online terpercaya yang membahas enkapsulasi dalam pemrograman.

Pada tahap awal, dilakukan permintaan kata kunci seperti "enkapsulasi", "pemrograman yang diarahkan pada objek", "kontrol akses pengubah", "Java", dan kata kunci terkait lainnya. Pencarian sumber literatur kemudian dilakukan melalui basis data akademik dan sumber-sumber online terpercaya.

Selanjutnya dilakukan seleksi dan penilaian sumber literatur yang sesuai dengan topik. Sumber-sumber literatur yang relevan dipilih berdasarkan kualitas, relevansi, dan kelembutan. Metodologi penelitian yang digunakan dalam studi yang terkait juga dinilai untuk memastikan validitas informasi yang diperoleh.

Setelah itu, dilakukan analisis literatur terhadap sumber-sumber yang terpilih. Isi dari literatur tersebut dianalisis untuk mengidentifikasi konsep-konsep, manfaat, dan enkapsulasi implementasi dalam pemrograman berorientasi objek. Temuan-temuan yang muncul dari analisis literatur kemudian digunakan untuk menjelaskan konsep enkapsulasi dan kegunaan secara rinci.

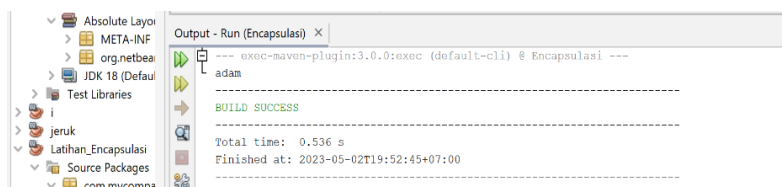
Penulisan artikel ini dilakukan dengan mengorganisir informasi dari sumber-sumber literatur yang telah dianalisis. Temuan-temuan dan argumen-argumen yang relevan disajikan dalam urutan yang logis dan terstruktur. Evaluasi dan sintesis informasi dilakukan untuk membentuk pemahaman yang komprehensif tentang enkapsulasi pemrograman dalam objek yang diarahkan.

Metode literature review ini memungkinkan kita untuk mendapatkan pemahaman yang mendalam tentang konsep enkapsulasi dalam pemrograman berorientasi objek serta gunanya dalam pemeliharaan kode.

IV. PEMBAHASAN

Contoh Enkapsulasi padapemograman Java :

Gambar 1.koding

Gambar 1 menunjukkan kode program dalam bahasa Java. Kode tersebut mendefinisikan sebuah kelas bernama "Orang" yang menerapkan konsep enkapsulasi. Kelas ini memiliki atribut "nama" yang dinyatakan sebagai private, yang berarti hanya dapat diakses secara langsung dari dalam kelas "Orang" itu sendiri. Konsep enkapsulasi ini membantu melindungi integritas data dan mencegah akses tidak sah atau modifikasi langsung terhadap atribut tersebut dari luar kelas.

Selain atribut "nama", kelas "Orang" juga memiliki dua metode, yaitu "getNama()" dan "setNama(String newNama)". Metode "getNama()" digunakan untuk mengembalikan nilai dari atribut "nama", sementara metode "setNama(String newNama)" digunakan untuk mengubah nilai atribut "nama" dengan nilai baruyang diberikan sebagai parameter.

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.encapsulasi;

/**
 *
 * @author ASUS
 */
public class orang {
    private String nama;

     public String getNama(){
      return nama;
     }
    public void setnama(String newNama) {
        this.nama=newNama;
}
}
```

Gambar 2.koding

Gambar 2 menunjukkan program bagian "main". Di sini, kita membuat objek "obj1" dari kelas "Orang" menggunakan konstruktor "orang baru()". Objek "obj1" merepresentasikan individu yang memiliki atribut "nama". Kemudian, kita menggunakan metode "setNama()" dari objek "obj1" untuk mengubah nilai atribut "nama" menjadi "adam".

Gambar 3.hasil

Gambar 3 menunjukkan hasil output dari program ketika dijalankan. Output yang dihasilkan adalah "adam". Hal ini terjadi karena kita telah mengubah nilai atribut "nama" dari objek "obj1" menjadi "adam" menggunakan metode "setNama()". Metode ini memungkinkan kita untuk melakukan validasi atau mengolah data sebelum memperbarui nilai atribut, sehingga kita dapat memastikan bahwa aturan bisnis atau kebutuhan aplikasi terpenuhi.

Setelah mengubah nilai atribut "nama", kita menggunakan metode "getNama()" untuk mendapatkan nilai atribut "nama" yang telah diubah. Dalam contoh ini, nilai yang dikembalikan adalah "adam". Metode "getNama()" menyediakan akses yang disimpan untuk membaca nilai atribut "nama" tanpa memungkinkan perubahan langsung terhadap atribut tersebut.

Dengan menggunakan konsep enkapsulasi, kita dapat membatasi akses langsung ke atribut dalam kelas, sehingga memungkinkan kita melindungi data dan menerapkan logika bisnis yang relevan. Ini membantu dalam meminimalkan kesalahan akses yang tidak menguasai atau memproses data yang tidak sah.

Dengan demikian, melalui penjelasan dari ketiga gambar tersebut, kita dapat memahami bagaimana konsep enkapsulasi diimplementasikan dalam program Java dan bagaimana penggunaannya mengamankan akses terhadap atribut serta memungkinkan pemulihan yang diatur melalui metode-metode yang ditentukan.

V. KESIMPULAN

Dalam artikel ini, kami menggambarkan konsep enkapsulasi dalam pemrograman berorientasi objek dan pentingnya dalam pengembangan perangkat lunak. Kami menjelaskan bahwa enkapsulasi pertama kali diperkenalkan oleh bahasa pemrograman Simula pada tahun 1960-an dan kemudian diadopsi oleh bahasa pemrograman lain seperti Smalltalk, C++, dan Java.

Enkapsulasi memungkinkan pengembang perangkat lunak untuk menyembunyikan detail implementasi dari penggunaan objek, sehingga objek dapat digunakan dengan lebih mudah dan aman. Ini memberikan fleksibilitas bagi pengembang untuk mengubah implementasi objek tanpa mempengaruhi kode pengguna, dan juga membantu mencegah perubahan yang tidak merugikan objek data oleh pengguna kode, yang dapat menyebabkan kerusakan sistem.

Metode yang digunakan dalam penulisan artikel ini adalah refleksi pustaka, di mana kami mencari dan menganalisis sumber-sumber literatur yang relevan tentang enkapsulasi dalam pemrograman. Tinjauan pustaka memungkinkan kami untuk mendapatkan pemahaman yang mendalam tentang konsep enkapsulasi dan manfaatnya dalam memelihara kode.

Dengan memahami konsep enkapsulasi dan menerapkannya dengan baik, perangkat lunak pengembang perangkat lunak dapat meningkatkan keamanan, perbaikan, dan efisiensi kode mereka. Enkapsulasi merupakan prinsip penting dalam pemrograman berorientasi objek yang membantu menciptakan perangkat lunak yang lebih terstruktur, mudah dipahami, dan mudah dikembangka

DAFTAR PUSTAKA

Buchmann, A. (2016). Konstruksi Perangkat Lunak Berorientasi Objek. Balai Prentice.

Didik Evaristus, "Encapsulation – School of Information Systems," 2017. https://sis.binus.ac.id/2017/09/04/encapsulation/ (diakses 29 Mei 2023).

Deitel, P., & Deitel, H. (2017). Java: Cara Memprogram, Objek Awal (Edisi ke-11). Pearson.

[1]Herbert Schildt, *Java: Panduan Pemula, Edisi Kedelapan, Edisi ke-8*. 2018. Diakses: 29 Mei 2023. [Daring]. Tersedia pada: https://learning.oreilly.com/library/view/java-a-beginners/9781260440225/

Horstmann, CS, & Cornell, G. (2013). Core Java Volume I--Fundamentals (Edisi ke-10). Balai Prentice.

Deitel, P., & Deitel, H. (2017). Java: Cara Memprogram, Objek Awal (Edisi ke-11). Pearson.

Potamianos, A., & Tsoukalas, IA (2020). Enkapsulasi dan Penyembunyian Informasi dalam Bahasa Pemrograman Berorientasi Objek: Tinjauan Literatur Sistematis. Jurnal Sistem dan Perangkat Lunak, 161, 110510.

Hemmati, H., & Ramesh, B. (2018). Menyelidiki Dampak Enkapsulasi pada Kualitas Perangkat Lunak: Tinjauan Literatur Sistematis. Teknologi Informasi dan Perangkat Lunak, 99, 10-